Mozilla Security Review

General Information

Project Name:	MathJax WordPress Plugin
Security Reviewer:	Matthew Fuller (mfuller)
Date of Review:	6/27 - 6/29
Bugzilla Bug #:	694700

Background Information:

MathJax is a WordPress plugin that enables in-browser support for math-based equations (native text rather than pictures of formulas). It does so via JavaScript libraries and special font-families that are loaded onto the page when needed. The Metrics team has requested to enable the plugin on their blog.

From the MathJax Plugin Page:
"Mathjax enables enables rendering of embedded latex or mathml in HTML pages. This plugin adds this functionality to wordpress. The mathjax javascript is inject on-demand only to those pages which require it. This ensures that mathjax is not loaded for all pages, which will otherwise slow loading down.
The MathJax javascript can be delivered from your own server, or you can utilise the [MathJax Content Distribution Network (CDN)] (http://www.mathjax.org/docs/1.1/start.html#mathjax-cdn), which is the preferred mechanism as it offers increased speed and stability over hosting the Javascript and configuring the library yourself. Use of the CDN is governed by these Terms of Service.
You may embed latex using a variety of different syntaxes. The shortcode (http://codex.wordpress.org/Shortcode_API) syntax is preferred. So [latex]E=mc^2[/latex] will work out of the box. This also forces loading of mathjax.
Additionally, you can use native mathjax syntax -- $$E=mc^2$$ or (E=mc^2). However, if this is the only syntax used, the plugin must be explicitly told to load mathjax for the current page. This can be achieved by adding a [mathjax] shortcode anywhere in the post. For posts with both [latex]x[/latex] and $$x$$ syntaxes this is unnecessary.
You can use wp-latex syntax, $latex E=mc^2$. Parameters can be specified as with wp-latex but will be ignored. This means that mathjax-latex should be a drop in replacement for wp-latex.
You can also specify [nomathjax] -- this will block mathjax on the current page, regardless of other tags."

Helpful URLs:

Metrics Team Blog: http://blog.mozilla.org/metrics/
MathJax Plugin Page: http://wordpress.org/extend/plugins/mathjax-latex/
MathJax Home: http://www.mathjax.org/resources/faqs/
More Installation Instructions: http://dist.mathjax.org/mathjax/1.1-beta/docs/html/installation.html
2.0 Documentation: http://www.mathjax.org/docs/2.0/installation.html

Access Information:
How is the application accessed?
Is the application internal or publicly available?
If it is internal, what mechanism prevents non-members from accessing it?

The plugin is not “accessed” but rather “utilized” when a user loads a page with predefined MathJax specific text. The blog is hosted publicly on a mozilla.org domain.

Infrastructure and Backends:
What languages do the applications use?
What database language is used if applicable?
Are the running versions up to date?
What server is it running on?

The application installs as a WordPress plugin and uses JavaScript and font-families to accomplish its formula editing.

The content (JavaScript libraries and font faces) can either be hosted locally (preferred) or loaded through the MathJax CDN. The later option introduces additional cross-domain scripting security issues.

To display, the following bit of code must be added to every page using MathJax functions:

```
<script type="text/x-mathjax-config">
  MathJax.Hub.Config({
       extensions: ["tex2jax.js"],
       jax: ["input/TeX","output/HTML-CSS"],
       tex2jax: {inlineMath: [["$","$"],["\\(","\\)"]]}
  });
</script>
<script src="http://10.250.2.190/wp-content/plugins/mathjax-latex/MathJax.js"
type="text/javascript"></script>
```

To display a formula, a sample code:

```
<p>
\[
\frac{-b\pm\sqrt{b^2-4ac}}{2a}
\]
</p>
```

Accounts and Passwords:
If the mechanism to prevent general access is a password, how is the signup process handled?
How is account information stored?
Are passwords properly stored within databases if applicable?
Is a password policy in place?
Are accounts locked-out after a number of invalid logins?
How long are session cookies stored?

No accounts are used with this plugin.

Third-Party Resources:
Are third-party resources used (i.e. JavaScript libraries, images, CSS, etc.)?
Can those resources be trusted / are they from reputable sources?

Is there a chance the resource could be compromised?
Is it possible to host the resources locally to mitigate risks?
Is a third-party responsible for storage of user data?

Third-party resources can be used (the MathJax CDN) or they can be hosted locally. The scripts are “inject on-demand” meaning they are only loaded when called rather than on every page. No user-data is collected or stored via any resources.

The CDN offers an https version which would mitigate compromised resources, but the prefered option remains local hosting.

Data Handling:
What kind of data is transferred between the user and the application?
Is this data generated by the user or generated automatically?
Can the data be trusted?
How is the data sent to the application (i.e. JSON format, plain-text, GET/POST, etc.)?
How is the data handled by the server/application?
Can the data be manipulated in transit?
What happens if the data is altered?
What is done with the data once it is received (i.e. stored in a database, displayed to users)?
Is any data storage done via cookies? If so, what kind of data is stored via this method?

There is no data transfer beyond the original text on the page. Instead, the plugin is merely loading an additional font face that is required to properly display math formulas. The user is not submitting data (besides comments, which are already checked for invalid parameters), the author of the blog post is posting the data with additional tags that call the MathJax renderer to display that text differently. Unless the admin console of WordPress were compromised, there is no user-interaction beyond the post author.

Data Sensitivity:
What kind of data is being stored and/or manipulated by the application?
Does this data need to be encrypted in transit? In storage?
What is the impact if this data is lost/stolen?

The data is not sensitive. The affected data are math formulas posted on a public-facing blog by the metrics team.

Application Administration:
Is there an administration console?
Can it be accessed publicly?
How is it secured if so?
Are correct methods used to prevent admin actions from being performed outside of the admin console (i.e. using CSRF tokens)?
Are there any configuration pages that should not be made public?

There is no administration console. The only files used are JavaScript, HTML, and font-families saved within a directory (most likely wp-content/plugins/mathjax-latex).

There are configuration pages, but they are merely “test” pages showing that, yes, MathJax is working and samples of how formulas are written. They do not contain sensitive data.

Security Coding Flaws:
Have all user inputs been sanitized?
Do all URL variables pass through sanitization?
Is data from databases escaped properly?
Are CSRF tokens used to prevent POSTs from outside websites?
If a database is used, are protections against SQL injection in place?

User input is only allowed via comments on the blog post page. The sanitization of the comments is not handled by this plugin, but by WordPress. This comment does enable additional interpretation of input by converting it to MathJax formula format, however this does not supercede the prior sanitization of input.

Testing:
List all tests performed on the application

1. Browsed to URL hosting MathJax
2. Reviewed all files, looking for non-JS, CSS, HTML, etc.
3. Looked for input variables (?var=)
4. Fuzzed above variable(s)
5. Are user comments interpreted with MathJax?
6. Can user comments inject scripts to trick MathJax into executing JS?

Results:
List the results of the above tests

1. Directory contents displayed
2. Found one conf.py
3. wp-content/plugins/mathjax-latex/MathJax.js?config=TeX-AMS-MML_HTMLorMML
4. No alerts()
5. Yes
6. Doesn't appear so.

Meetings and Notes:

Note: the WordPress plugins page currently has version 1.1 listed for download. However, the MathJax website lists version 2.0 as being the current version. It can be installed from their site directly. This review is being conducted for version 2.0.

Note: Plugin seems to not block directory listing by default. This should be enabled.

Note: conf.py in one directory, make inaccessible.